KPM202 系列用户手册

版本号：V1.0

河南康派

版本修订

版本号	修订日期	描述	审核
V1.0	20160907	创建文档	

特别说明

目录

一、产品概述

LT8013 是一款基于 RISC 架构五级流水线的芯片作为主处理器的嵌入式计算机。该 CPU 是以 ARM926EJS 为核心的系统级单芯片，内置 64MB DDR2，提供加解密软件，提供 AES、DES/3DES 加解密，最高支持 300MHz 的频率。系统提供有线网络通讯，同时也提供无线 GPRS 通讯，具有体积小、功耗低、效率高等特点，适用于电力集中器、HMI、工业控制、网关等场合。

二、产品特性

1. 硬件特性

- NUC970 CPU：
  - 32bit ARM926EJ-S，主频 300MHz，1.1MIPS/MHz，最高支持 300MHz
  - 16KB I-cache，16KB D-cache
  - 支持 MMU，支持 JTAG Debug

- 内存：
  - 内置 64Mbyte DDR2，56Kbyte SRAM
  - 高达 150MHz 的 SDRAM 时钟

- FLASH：
  - 128Mbyte NANDFlash，最大支持 512Mbyte（可定制 128M/256M/512M）
  - 支持 SLC、MLC 等类型的 NAND FLASH

- 加密：
  - 支持 PRNG/DES/3DES/AES/SHA/HMAC 加密，最高 256 位加密模式

- 看门狗：
  - 内置 WDT，溢出时间小于 14 秒，支持空闲唤醒和掉电唤醒

- RTC:
  - 实时时钟，内置供电电池

- RS232：
  - 1 路 RS232 通讯端口，内置 ESD 保护，全隔离保护设计

- RS485：
  - 2 路 RS485 通讯端口，内置 ESD 捉住，全隔离保护设计

- 网络：
  - 1 路 10M/100M 自适应工业以太网，标准 RJ45 接口
  - 15KV TVS 保护

- GPRS：
  - 射频波段 850/900/1800/1900MHz
  - 2 个 SIM 卡接口（并接），1 个天线接口

- 电源：
  - 输入电压 6~24VDC，推荐使用 12VDC
  - 功耗

- 机械特性
  - 外壳金属材质
  - 尺寸：
  - 防护等级：

- 工作环境
  - 工作温度：
  - 工作温度：
  - 存储温度：

2. 软件特性

2.1 系统概述

Linux 是一个成熟而稳定的网络操作系统。将 Linux 植入嵌入式设备具有众多的优点。首先，Linux 的源代码是开放的，任何人都可以获取并修改，用之开发自己的产品。其次，Linux 是可以定制的，其系统内核最小只有约 134KB。一个带有中文系统和图形用户界面的核心程序也可以做到不足 1MB，并且同样稳定。另外，它和多数 Unix 系统兼容，应用程序的开发和移植相当容易。同时，由于具有良好的可移植性，人们已成功使 Linux 运行于数百种硬件平台之上。

2.2 Linux 作为嵌入式操作系统的主要优势

1）可应用于多种硬件平台。Linux 已经被移植到多种硬件平台，这对于经费，时间受限制的研究与开发项目是很有吸引力的。原型可以在标准平台上开发后移植到具体的硬件上，加快了软件与硬件的开发过程。Linux 采用一个统一的框架对硬件进行配置，不需要任何的许可或商家的合作关系，源代码可以免费得到。这使得采用 Linux 作为操作系统不会遇到任何关于版权的纠纷。毫无疑问，这会节省大量的开发费用。本身内置网络支持，而目前嵌入式系统对网络支持要求越来越高。Linux 的调试模块化使添加部件非常容易。

2）Linux 是一个和 Unix 相似、以内核为基础的、具有完全的内存访问管制，支持大量硬件（包括 X86、Alpha、ARM 和 Motorola 等现有的大部分芯片）等特性的一种通用操作系统。其程序源码全部公开，任何人可以修改并在 GUN 通用公共许可证（GNU General Public Licence）下发行。这样，开发人员可以对操作系统进行定制，适应其特殊的需要。

3）Linux 带有 Unix 用户熟悉的完善的开发工具，几乎所有的 Unix 系统的应用软件都已移植到了 Linux 上。Linux 还提供了强大的网络功能，有多种可选择窗口管理器（XWindows）。其强大的语言编译器 GCC，C++等也可以很容易得到，不但成熟完善，而且使用方便。

2.3 LT8103 系统特性

LT8103 预装基于 nuvoton NUV970 的 Linux 操作系统，版本为 3.10.101。满足 POSIX 标准或类 UNIX 平台的应用程序。针对系统特有的硬件设备，内核提供了简单、易用的驱动接口，可加速用户的应用程序开发。

LT8103 系统的软件系统共分为 3 部分，分别为 Bootloader、linux 内核和 rootfs。Bootloader 是遵循 GPL 条款的开放源码项目，UBoot 主要是引导内核的启动，支持 NFS 挂载、NAND Flash 启动；linux 内核是整个操作系统的最底层，负责整个硬件的驱动，以及提供各种系统所需的核心功能；rootfs 是用于明确磁盘或分区上的文件的方法和数据结构，即在磁盘上组织文件的方法。

三、接口定义

1. 电源接口（VIN）

编号	标识符	功能说明
1	FG+	屏蔽地、保护地，可不接
2	—	系统电源地
3	+	系统电源，输入电压范围 DC6~24V，推荐使用 DC12V

2. SIM 卡接口（SIM CARD）

编号	标识符	功能说明
1	SIM CARD	2G、3G 的 SIM 卡接口，支持移动、联通卡

3. 天线接口（ANT）

编号	标识符	功能说明
1	GPRS	2G、3G 的 SMA 天线接口

4. 网络接口（ETH）

编号	标识符	功能说明
1	E0_TX+	以太网 ETH0_TX+
2	E0_TX-	以太网 ETH0_TX-
3	E0_RX+	以太网 ETH0_RX+
4	NC	未定义
5	NC	未定义
6	E0_RX-	以太网 ETH0_RX-
7	NC	未定义
8	NC	未定义

5. RS485 接口（RS485）

编号	标识符	功能说明
1	1A	第一通道 RS485 通讯 A 端口
2	1B	第一通道 RS485 通讯 B 端口
3	G	通讯系统地
4	2A	第二通道 RS485 通讯 A 端口
5	2B	第二通道 RS485 通讯 B 端口

6. RS232 接口（RS232）

编号	标识符	功能说明
1	R3	RS232 通讯 RX 端口
2	T3	RS232 通讯 TX 端口
3	G	通讯系统地

7. 指示灯

编号	标识符	功能说明
1	POW/RUN	一个为系统电源指示灯，一个为运行状态指示灯
2	RX1/TX1	第一通道 RS485 通讯指示灯
3	RX2/TX2	第二通道 RS485 通讯指示灯
4	RX3/TX3	RS232 通讯指示灯
5	STATUE/VBAT	GPRS 工作指示灯，一个为运行状态指示灯，一个为网络状态指示灯

四、系统软件

1. BootLoader 基本操作

1.1 简介

Bootloader 使用 u-boot，主要功能：

- 参数设置管理
- 通过串口或网络下载文件到 flash 或者 ram
- 启动 linux 操作系统
- 查询 CPU、内存等系统参数
- 内存读写

1.2 u-boot 主要命令

1.2.1 参数设置

1.2.1.1 查询参数

printenv

1.2.1.2 设置参数

setenv

如：

setenv ipaddr 192.168.1.177　　（设置 IP 地址）

setenv serverip 192.168.1.95　　（设置 tftp server 的 IP 地址，即本机 IP）

1.2.1.3 保存参数

saveenv

参数设置完毕后，只是保存在内存中，需要此操作将配置的参数保存到 Flash 中，保存完毕后，重启后才能生效。

1.2.2 网络命令

1.2.2.1 测试网络连接

ping

```
U-Boot> ping 192.168.1.1
Using emac device
host 192.168.1.1 is alive
U-Boot>
```

1.2.2.2 连接 TFTP Server

tftp addr filename

从 TFTP Server 下载指定文件 filename 到指定内存开始地址为 addr 开始的连续空间中

如：

tftp uImage

从 TFTP Server 下载文件 uImage 到内存地址为 0x100000(缺省时使用默认地址 0x100000)开始的连续空间中

```
U-Boot> tftp uImage
Using emac device
TFTP from server 192.168.1.95; our IP address is 192.168.1.177
Filename 'uImage'.
Load address: 0x100000
Loading: ################################################################
         ################################################################
         ################################################################
         ################################################################
         ################################################################
         ################################################################
         ##########################################################
         238.3 KiB/s
done
Bytes transferred = 2287928 (22e938 hex)
```

1.2.3 nandflash 命令

1.2.3.1 删除命令

nand erase start len

删除指定的开始地址，长度为指定长度的 nandflash 空间

如：

nand erase 200000 400000

删除从地址为 0x200000 开始的空间，删除长度为 0x400000

注：长度必需为 0x20000 的整数倍

1.2.3.2 写入命令

nand write source start len

将指定的内存开始地址，长度为指定长度的内容写入到指定地址的 nandflash 空间

如：

nand write 100000 200000 280000

将内存中起始地址为 0x100000 的内容写至 nandflash 起始地址为 0x200000 的空间，写入长度为 0x280000

注：长度必需为 0x20000 的整数倍

1.2.4 启动命令

bootm

启动 linux 操作系统

如：

bootm 100000

启动内存地址 0x100000 开始的 linux 内核（前提是 linux 内核已经下载到开始地址为 0x100000 的 SDRAM 中）

```
U-Boot> bootm 100000
## Booting kernel from Legacy Image at 00100000 ...
   Image Name:   Linux-3.10.101
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    2287864 Bytes = 2.2 MiB
   Load Address: 00008000
   Entry Point:  00008000
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...
```

1.2.5 复位命令

reset

复位系统，重新启动 u-boot

1.2.6 帮助

help

显示帮助信息

2. Linux 内核

LT8103 系列使用标准的嵌入式 Linux 系统，能够满足 POSIX 标准或类 UNIX

平台的应用程序移植到本系统。系统是主要特点：

- 支持完整的 TCP/IP 协议，支持 PPP 协议
- 支持 Telnet，为用户提供在本地计算机上完成远程主机工作的能力
- 支持 System V IPC
- 支持共享内存
- 软件支持浮点运算
- 支持 Memory Technology Device 技术，采用 NandFlash 作为系统存储介质
- 文件系统：
  - Ext2 linux 文件系统
  - NFS 网络文件系统
  - udev 设备文件系统
  - Proc 内核文件系统
  - Fat dos 文件系统
  - UBI 文件系统
- 设备驱动
  - USB 驱动
  - 网络驱动
  - UART 串口驱动
  - RTC 实时时钟
  - 看门狗（WTD）驱动
  - SD/MMC 卡驱动
  - IIC 驱动

3. 文件系统

文件系统是基于被划分的存储设备上的逻辑上单位上的一种定义文件的命名、存储、组织及取出的方法。如果一个 Linux 没有根文件系统，它是不能被正确的启动的。因此，我们需要为 Linux 创建根文件系统，并将其存储在 NAND FLASH 上，本系统采用 UBI 文件系统。

3.1 Linux 的根文件系统常用目录

3.1.1 /bin(binary)：包含着所有的标准命令和应用程序

3.1.2 /dev(device)：包含外设的文件接口，在 Linux 下，文件和设备采用同样的

方法访问，系统上的每个设备都在/dev 里有一个对应的设备文件

3.1.3 /etc(etcetera)：目录包含系统设置文件和其他的系统文件，例如/etc/fstab(file system table)记录了启动时要 mount 的 filesystem

3.1.4 /home：普通用户主目录

3.1.5 /lib(library)：存放系统最基本的库文件

3.1.6 /mnt：用户临时挂载文件系统的地方

3.1.7 /proc：Linux 提供的一个虚拟系统，系统启动时在内存中产生，用户可以直接通过访问这些文件来获得系统信息

3.1.8 /root：超级用户主目录

3.1.9 /sbin：存放系统管理程序，如 fsck、mount 等

3.1.10 /tmp(temporary)：存放不同的程序执行时产生的临时文件

3.1.11 /usr(user)：存放用户应用程序和文件

3.1.12 /program：存放外设的操作脚本与测试实例

3.2 BusyBox

BusyBox 最初是由 Bruce Perens 在 1996 年为 Debian GNU/Linux 安装盘编写的。其目标是在一张软盘上创建一个可引导的 GNU/Linux 系统，这可以用作安装盘和急救盘。一个张软盘可以保存大约 1.4-1.7MB 的内容，因此这里没有多少空间留给 Linux 内核以及相关的用户应用程序使用。

BusyBox 揭露了这样一个事实：很多标准 Linux 工具都可以共享很多共同的元素。例如，很多基于文件的工具（比如 grep 和 find）都需要在目录中搜索文件的代码。当这些工具被合并到一个可执行程序中时，它们就可以共享这些相同的元素，这样就可以产生更小的可执行程序。实际上，BusyBox 可以将大约 3.5MB 的工具包装成约 200KB 大小。这就为可引导的磁盘和使用 Linux 的嵌入式设备提供了更多功能。

采用 BusyBox 是缩小根文件系统的好办法，因为其中提供了系统的许多基本指令，且其体积很小。众所周知，瑞士军刀以其小巧轻便、功能众多而闻名世界，成为各国军人的必备工具，并广泛应用于民间，而 BusyBox 也被称为嵌入式 Linux 领域的“瑞士军刀”。

3.3 LT8103 主要命令

LT8103 采用 BusyBox 以减少系统占用资源，经裁减，保留了一些主要的和常用的命令。

3.3.1 文件列表

ls

查看当前目录

ls+目录名称(包括路径)

查看指定目录名称中的文件

使用方法：

ls　　//查看当前目录

ls /home/user　　//查看 user 中的文件

3.3.2 更换当前目录

cd

使用方法：

cd　dir　　//更换到当前目录下的 dir 目录

cd　/　　//更换到根目录

cd　..　　//切换到上一级目录

3.3.3 复制

cp

使用方法：

cp　src　des　　//把文件 src 复制为 des

cp　/root/src　./　　//把 root 下的文件 src 复制到当前目录

cp -av src_dir des_dir　　//把目录 src_dir 复制为 des_dir，两目录内容一样

cp –fr src_dir des_dir　　//将整个目录非链接方式复制，当 src_dir 目录带有符号链接时，两个目录不相同

3.3.4 系统时间日期

date

使用方法：

date　　//显示当前系统日期时间

date –s 12:34:56　　//设置系统时间为 12:34:56

date –s 2010-01-02　　//设置系统日期为 2010-01-02

3.3.5 分布查看

more

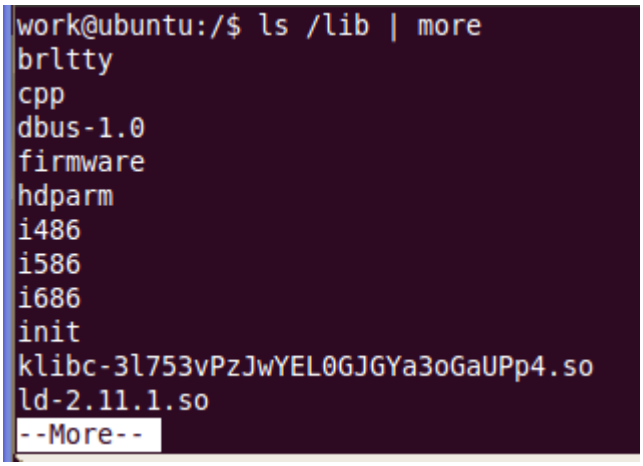
使用方法：

more　　//分页命令，一般通过管道将内容传给它

ls /filename | more　　//通过管道传给 more，分页查看目录 filename 下的

文件

对于文件较多的目录，使用上述命令分页查看，按回车键往下翻一行，按空格键往下翻一页，按 q 退出查看。如：



3.3.6 回显

echo

使用方法：

echo “message”　　//显示一串字符

echo “message message2”//显示不连续的字符串

3.3.7 挂接

mount

使用方法：

mount –t nfs 192.168.1.100:/nfs /mnt　//将 nfs 服务的共享目录/nfs 挂接到本地/mnt 目录

3.3.8 移动

mv

使用方法：

mv　src　des　　//将文件 src 更名为 des

mv　/file1/src　/file2/　//将 file1 下的 src 文件移动到 file2 目录下

3.3.9 删除

rm

使用方法：

rm　file_name　　//删除一个叫做 file_name 的文件

rm　-rf　dir　　//删除当前目录下名称为 dir 的整个目录

3.3.10 搜索

grep

使用方法：

grep –ir “chars” *　　//在当前目录下的所有文件中查找字符串 chars，并忽略大小定（-i），查找包括当前目录下的子目录（-r）。

3.3.11 查找

find

使用方法：

find /path –name file　　//在 path 目录下查找文件名为 file 的文件

3.3.12 查看文件内容

cat

使用方法：

cat file　　//显示文件 file 的内容(以 ASCII 码表示)

3.3.13 权限修改

chmod

使用方法：

chmod a+x file　　//把 file 文件设置为可执行，脚本类文件必须要这样设置，否则需要使用 bash file 才能执行。参数 a 可使所有用户具有对此文件的可执行权限。

例如对 keytest 文件添加可执行权限：

```
/program # ls -l
-rw-r--r--    1 root     0            9039 Mar 11 23:09 iotest
-rw-r--r--    1 root     0            7403 Mar 11 23:09 keytest
/program # chmod a+x keytest
/program # ls -l
-rw-r--r--    1 root     0            9039 Mar 11 23:09 iotest
-rwxr-xr-x    1 root     0            7403 Mar 11 23:09 keytest
/program #
```

类似的可以使用 chmod a+r file 和 chmod a+w file 命令分别设置文件的读和写权限。

3.3.14 压缩与解压

tar

解压使用方法：

tar -xjvf filename.tar.bz2　　//把文件 filename.tar.bz2 解压到当前目录下

tar -xzvf filename.tar.gz　　//把文件 filename.tar.gz 解压到当前目录下

压缩使用方法：

tar -cjvf filename.tar.bz2 /filename　　//把文件或者文件夹 filename 压缩为 filename.tar.bz2

tar –czvf filename.tar.gz /filename　　//把文件或者文件夹 filename 压缩为 filename.tar.gz

参数说明：

-x：解压压缩文件

-c：建立压缩文件

-j：解压或者压缩后缀名为.bz2 的文件

-z：解压或者压缩后缀名为.gz 的文件

-v：显示被解压或压缩的文件

-f：指定包的文件名

3.3.15 编辑

vi

使用方法：

vi file　　//编辑文件 file

在 vi 状态下，输入命令的方式为先按 ctrl+c，然后输入

:q　　//退出

:q!　　//退出不保存

:wq　　//保存退出

:w　　//写入文件

:r file　　//不询问方式写入文件

%s/oldchars/newchars/g　　//将所有字符串 oldchars 换成 newchars

3.3.16 创建节点

mknod

使用方法：

mknod /dev/tty1 c 4 1　　//创建字符设备 tty1，主设备号为 4，从设备号为 1，即第一个 tty 终端。

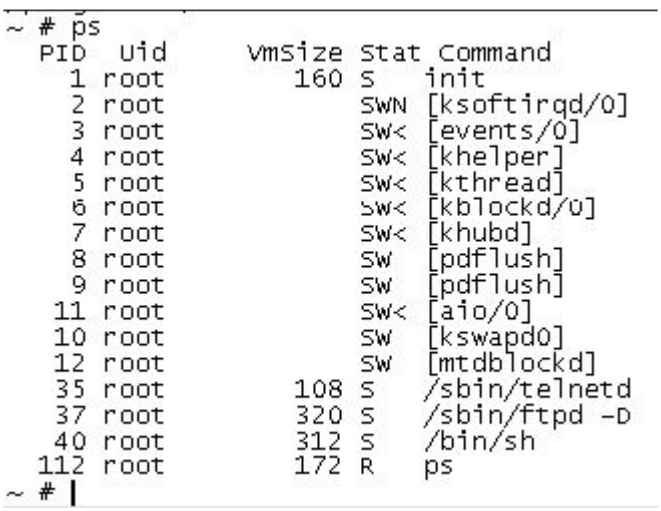
3.3.17 进程查看

ps

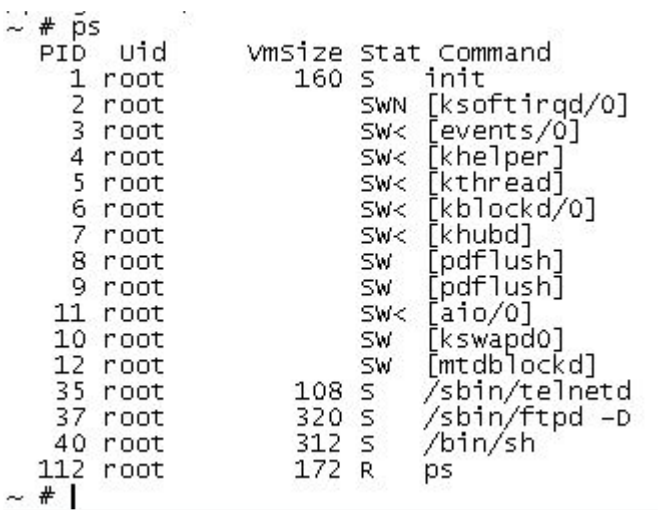
使用方法：

ps　　//显示当前系统进程信息

如下图所示：



ps –ef　　//显示系统所有进程信息



3.3.18 杀死进程

kill

使用方法：

kill -9 250　　//将里程编号为 250 的程序杀死

3.3.19 设置环境变量

export

使用方法：

export LC_ALL=zh_CN.GB2312　　//将环境变量 LC_ALL 的值设为 zh_CN.GB2312

3.3.20 启动信息显示

dmesg

使用方法：

dmesg　　//显示 kernel 启动及驱动装载信息

3.3.21 网络设置命令

ifconfig

使用方法：

ifconfig　　//查看网卡的状态，在开发板上执行时会显示网卡与本地回路(lo，loopback)的信息

ifconfig eth0 192.168.1.230 netmask 255.255.255.0　　//表示设置网卡 1 的地址为 192.168.1.230，掩码为 255.255.255.0，不写 netmask 参数则默认为 255.255.255.0

注：开发板有缺少的网络 IP，这是通过在启动脚本/etc/inin.d/rcS 文件中使用 ifconfig 实现的

ifconfig eth0 down　　//关闭网卡 1

ifconfig eth0 up　　//启动网卡 1

设置 MAC 地址：

ifconfig eth0 down

ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx

ifconfig eth0 up

其中 xx:xx:xx:xx:xx:xx 为设置的 MAC 地址，如 00:12:34:56:78:90，注意第一个字节必须为偶数

3.3.22 设置网关

route

使用方法：

route　　//显示当前路由设置情况

route add default gw 192.168.1.1　　//设置 192.168.1.1 为默认的路由

route del default　　//将默认的路由删除

route add –net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.1 dev eth0

3.3.23 测试网络连通

ping

使用方法：

ping –c 3 192.168.1.1　　//向 192.168.1.1 连续发送 3 次测试包，以验证网络是否连接正常。如果连接正常，则显示结果如下：

```
~ # ping -c 3 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: seq=0 ttl=64 time=0.967 ms
64 bytes from 192.168.1.1: seq=1 ttl=64 time=0.692 ms
64 bytes from 192.168.1.1: seq=2 ttl=64 time=2.700 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.692/1.453/2.700 ms
~ #
```

3.3.24 RTC 时钟命令

hwclock

使用方法：

hwlock　　//显示当前 RTC 的时间

hwclock –s　　//将当前 RTC 的时间同步到 Linux 系统时间

hwclock –w　　//将 Linux 系统时间同步到 RTC 的时间

3.3.25 系统复位命令

reboot

使用方法：

reboot　　//系统重新启动

五、应用软件

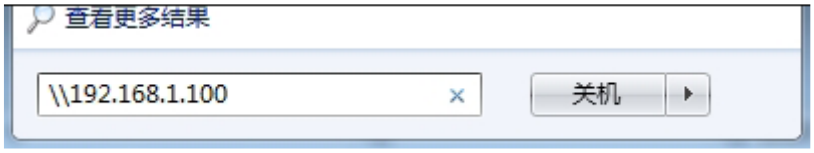
1. 开发环境

ubuntu 10.04，eclipse

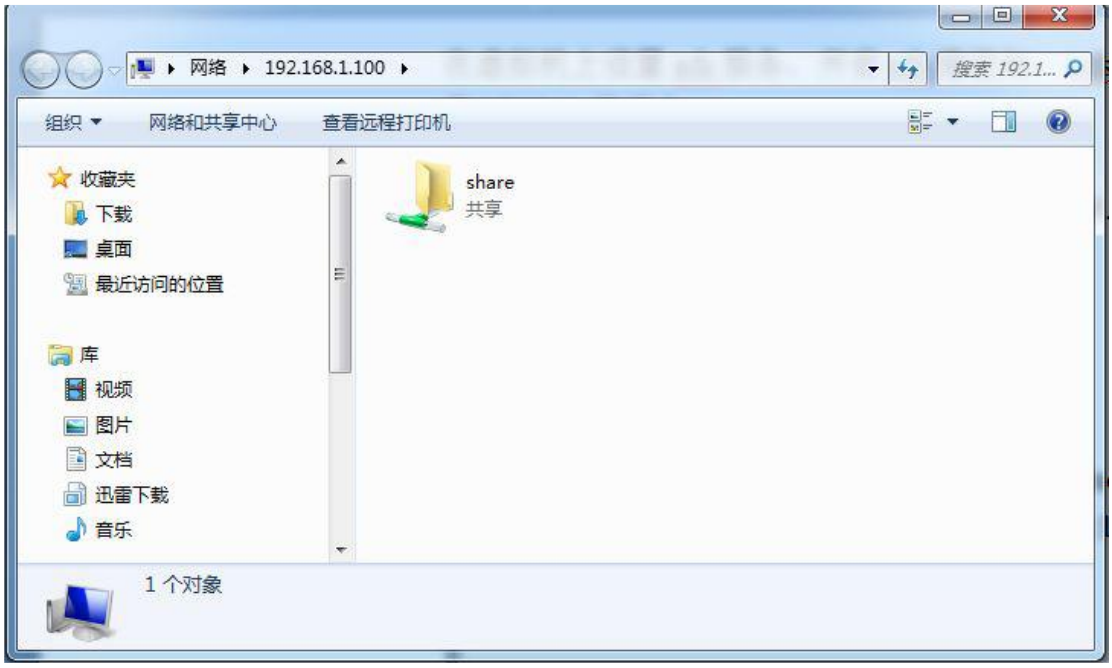
交叉编译工具：arm-linux-gnueabihf-gcc(版本 4.8.4)

2. 虚拟机与 windows 共享目录

2.1 随机光盘中的虚拟机开启了 samba 服务，可与 windows 共享目录，便于文件复制。在 windows 下的开始菜单中选择运行，输入\\192.168.1.101(此 IP 为虚拟机的 IP 地址)，然后回车



2.2 第一次使用需要验证登录信息。在弹出的窗口中输入密码，用户名和密码与登录虚拟机的一样。



3. 使用 NFS 网络文件系统

3.1 服务器

在虚拟机上设置 NFS 服务，开启 NFS 路径，如/nfs

3.2 终端设备

执行命令：mount –t nfs 192.168.1.100:/nfs /mnt –o nolock

/mnt 即为虚拟机的/nfs 目录内容（注：192.168.1.100 为虚拟机的 IP 地址）

4. 开机自启动

4.1 建立启动脚本

在目标板文件系统/program 下建立 startup.sh 文件，输入如下内容：

#! /bin/sh

/prog/myprog　　#启动的程序 myprog 和路径

4.2 修改属性

chmod +x startup.sh

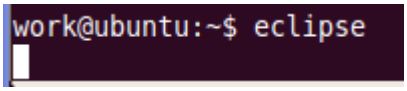
4.3 重启系统

reboot

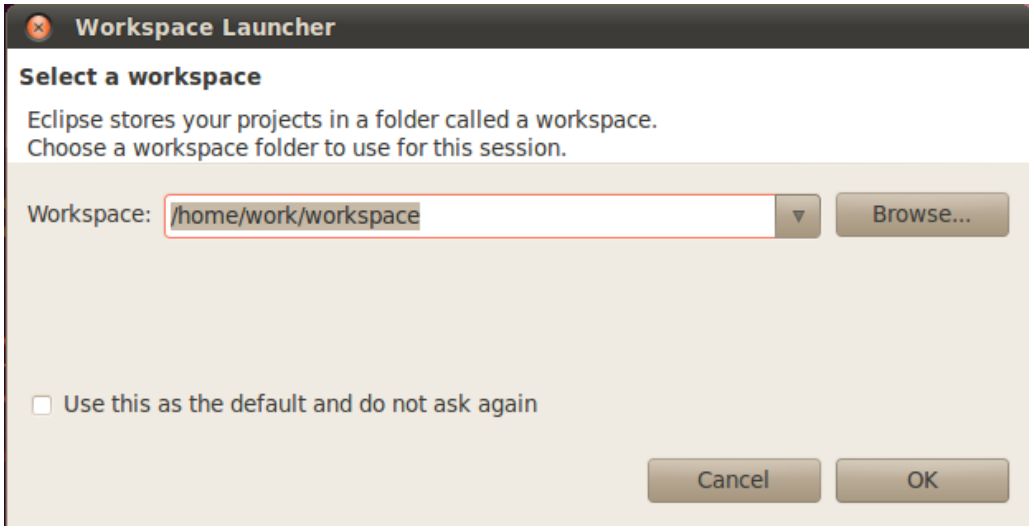
5. eclipse 使用说明

5.1 在虚拟机下建立一个简单的 helloworld 程序

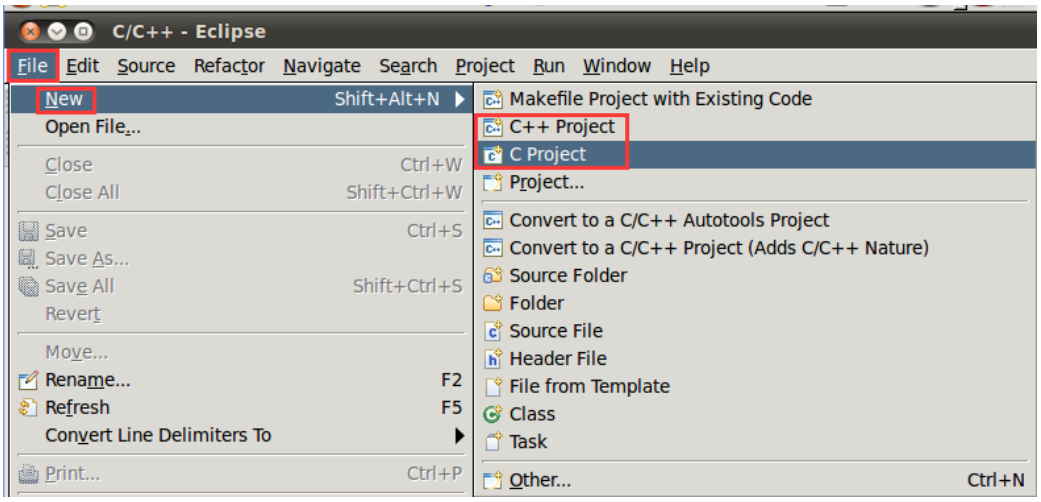
在终端输入：eclipse，等待编程软件启动



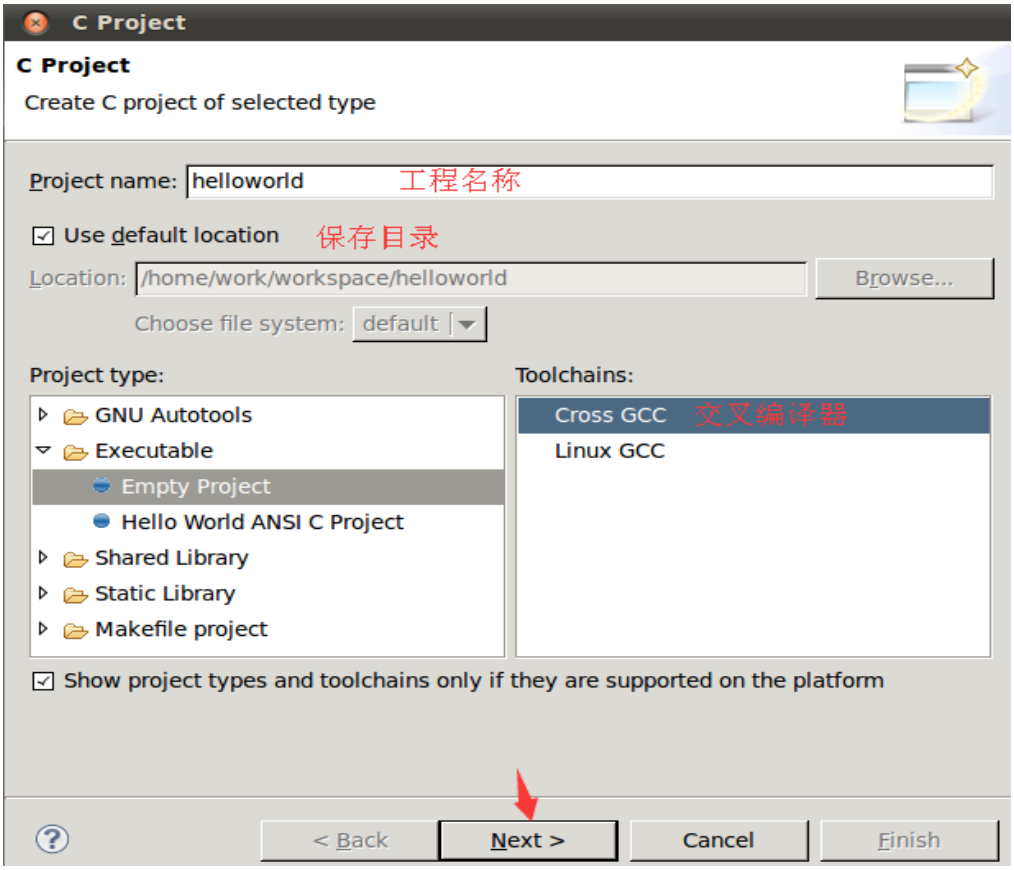
5.2 选择工作路径，可以使用默认目录，也可以自定义目录



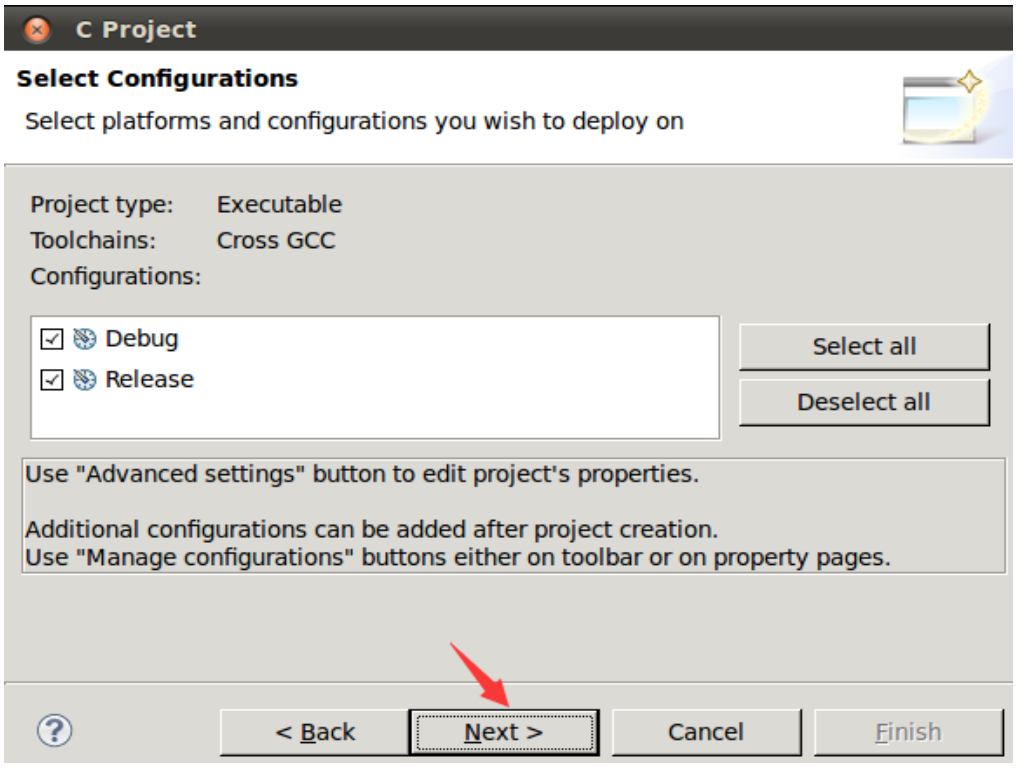
5.3 新建工程：File → New → c project(c++程序的选择 c++ project)，如图所示



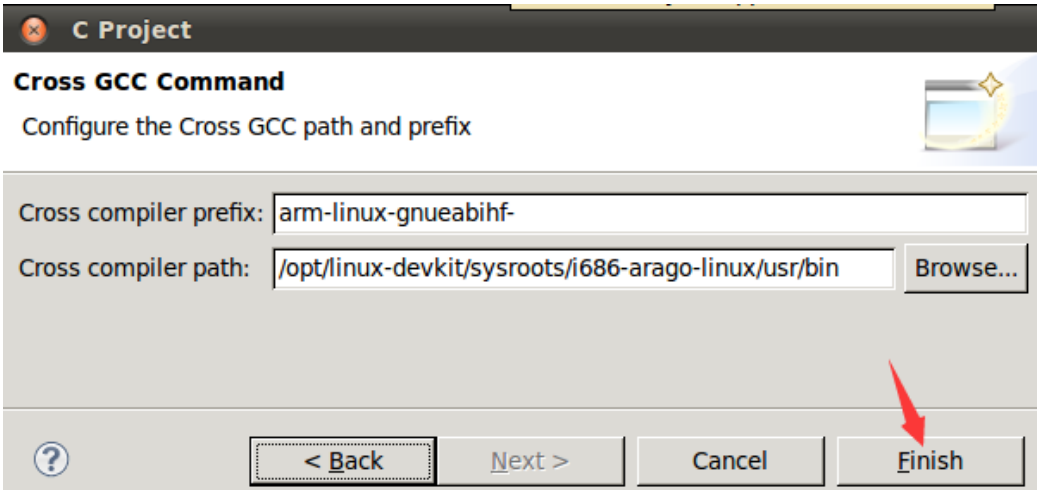
5.4 输入工程名称，选择保存路径，选择交叉编译，如下图所示，完成后进行下一步



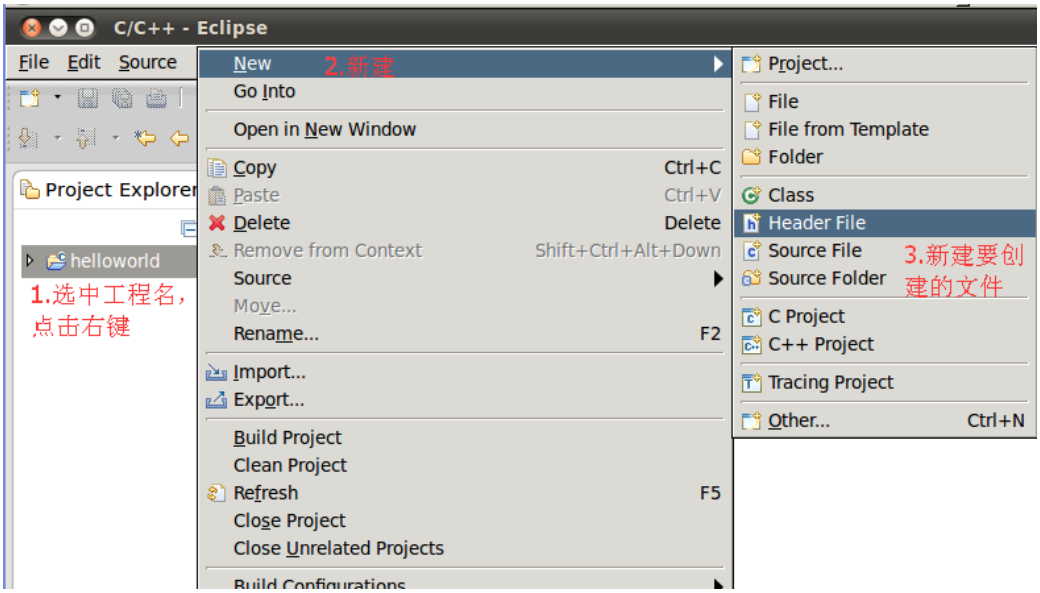
5.5 此步使用默认配置，继续下一步



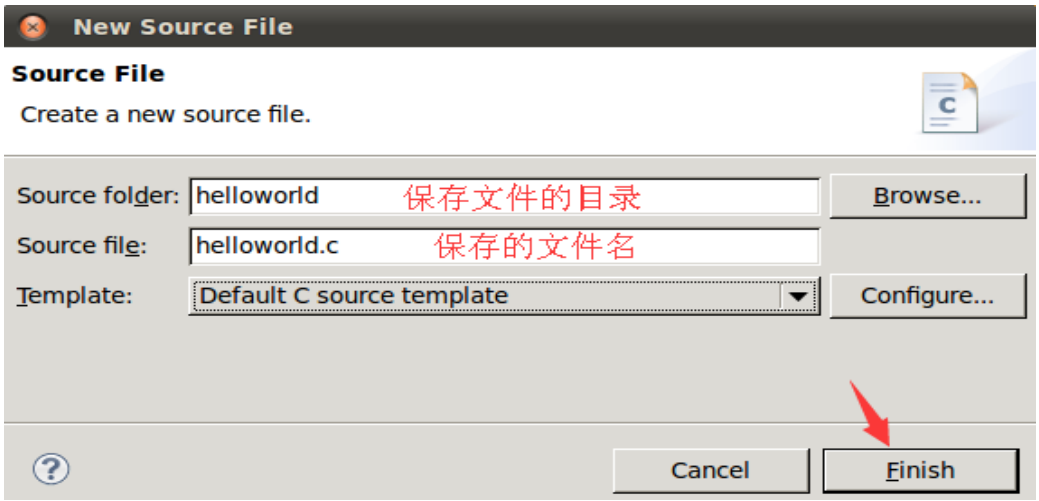
5.6 填写交叉编译器及路径，如下图所示（光盘提供了配置好的虚拟机系统，系统内包括了交叉编译器），填写完后选择 finish，完成工程创建



5.7 为工程添加文件



5.8 输入文件名，完成源文件的创建



5.9 编写代码

helloworld.c

```
/*
 * helloworld.c
 *
 *  Created on: Sep 12, 2016
 *      Author: work
 */
#include <stdio.h>

int main()
{
    printf("hello world\n\r");
    return 0;
}
```

5.10 保存文件，交叉编译，编译器版本为 4.8.4

```
work@ubuntu:~$ arm-linux-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gcc
COLLECT_LTO_WRAPPER=/opt/arm_linux_4.8/bin/../libexec/gcc/arm-nuvoton-linux-uclibceabi/4.8.4
/lto-wrapper
Target: arm-nuvoton-linux-uclibceabi
Configured with: ../configure --prefix=/usr/local/arm_linux_4.8 --target=arm-nuvoton-linux-u
clibceabi --enable-static --disable-__cxa_atexit --with-gnu-ld --disable-libssp --disable-mu
ltilib --enable-target-optspace --disable-libsanitizer --enable-tls --disable-libmudflap --e
nable-threads --without-isl --without-cloog --with-float=soft --disable-decimal-float --with
-abi=aapcs-linux --with-cpu=arm926ej-s --with-float=soft --with-mode=arm --enable-languages=
c,c++ --enable-poison-system-directories --enable-shared --disable-libgomp --with-sysroot=/u
sr/local/arm_linux_4.8 --with-build-time-tools=/usr/local/arm_linux_4.8/arm-nuvoton-linux-uc
libceabi/bin --enable-shared
Thread model: posix
gcc version 4.8.4 (GCC)
```

不显示任何提示信息表示编译通过

```
work@ubuntu:~/workspace/helloworld$ arm-linux-gcc -o helloworld helloworld.c
work@ubuntu:~/workspace/helloworld$
```

5.11 将编译好的文件拷贝到 nfs 目录（nfs server 服务器安装，请参照虚拟机设置相关文档）

```
work@ubuntu:~/workspace/helloworld$ cp helloworld /nfs/
work@ubuntu:~/workspace/helloworld$
```

5.12 在 LT8103 终端上挂载 nfs 网络文件系统（在重启前不需要再挂载）

```
~ # mount 192.168.1.104:/nfs /mnt/ -o nolock
~ #
```

192.168.1.104 为虚拟机 IP；/nfs 为虚拟上的目录；/mnt 为 LT8103 系统目录。

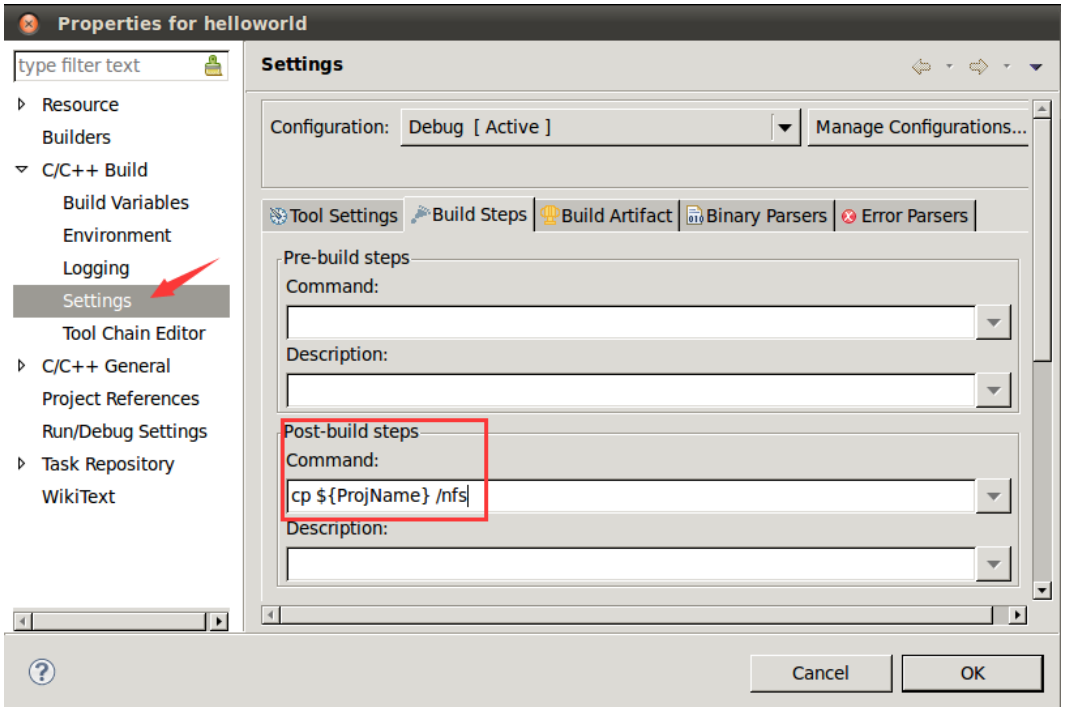
5.13 进行/mnt 目录，运行程序

```
/mnt # ./helloworld
helloworld
/mnt #
```

附 1：eclipse 参数配置说明

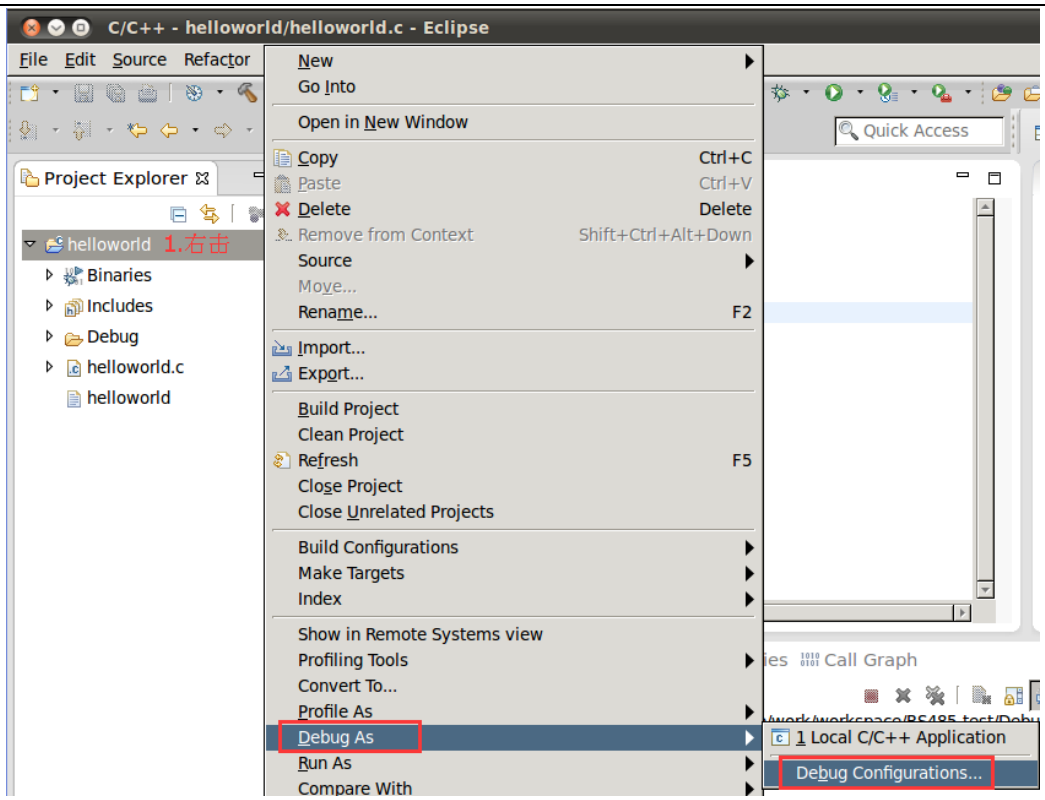
附 1.1 自动复制文件到指定目录下

选择 project 菜单 → Properties → C/C++ Build → Settings，选择选项卡 Build Steps，在 Post-build steps 的 Command 框中输入：cp ${projName} /nfs，OK 完成。



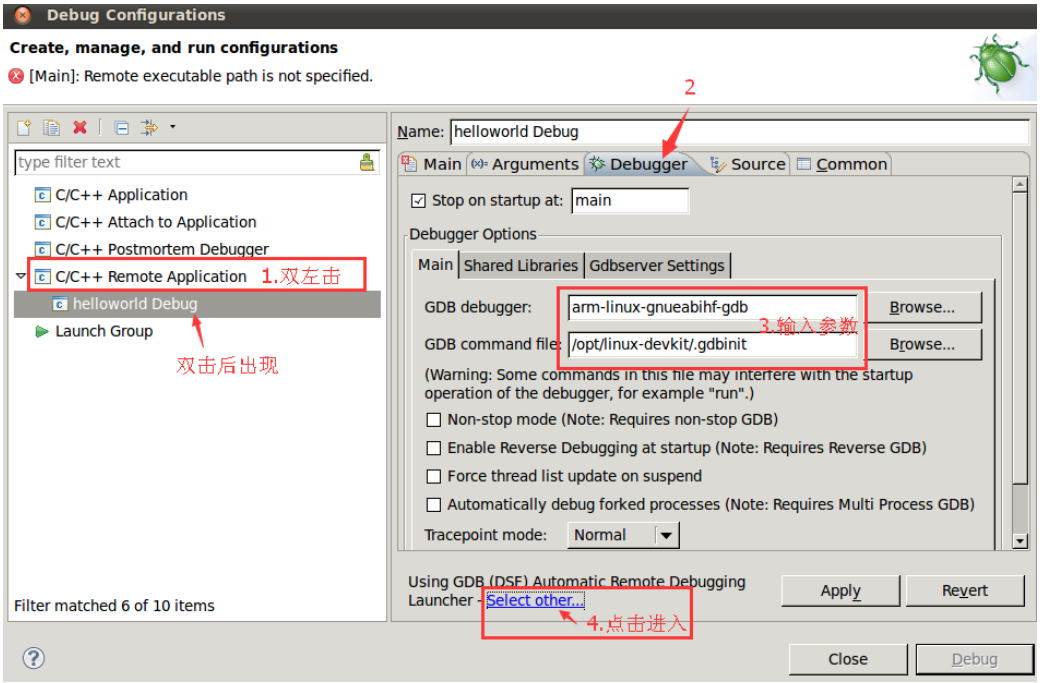
附 1.2 调试设置

在工程浏览器中，选中工程名，在右击菜单中选择 Debug As → Debug Configuration…

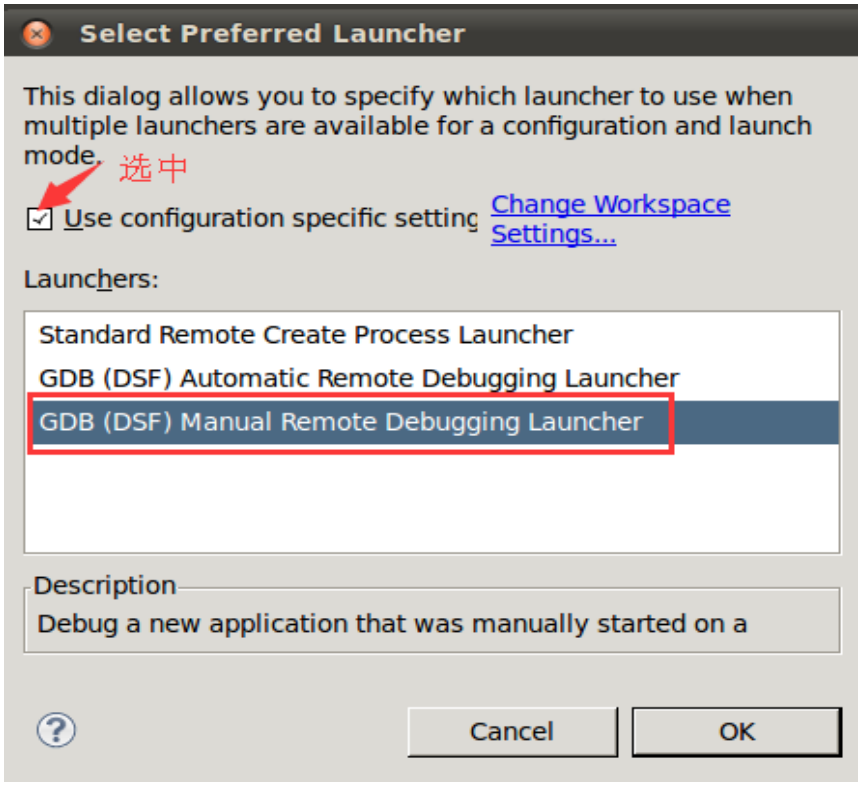


附 1.2.1 在打开的对话框中，按照以下图片步骤配置，其中第 3 步的参数分别为：

GDB debuger:：arm-linux-gnueabihf-gdb
GDB command file:：/opt/linux-devkit/.gdbinit



附 1.2.2 选择加载方式为手动加载

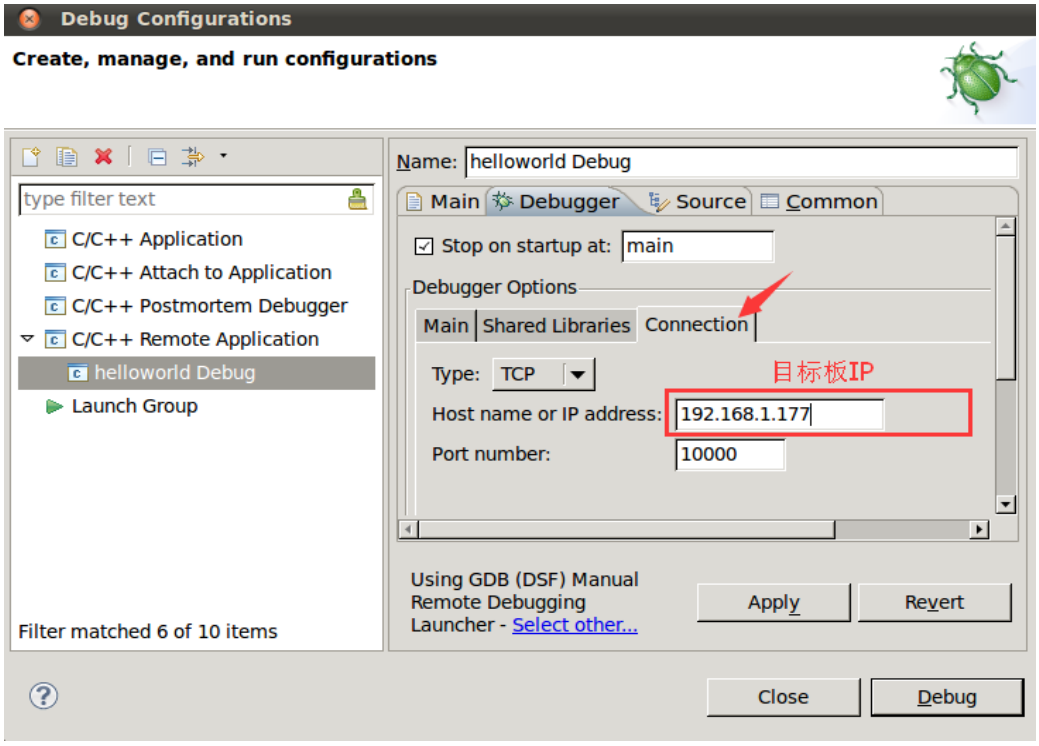


1.2.3 Connection 属性：

Host name or IP address：192.168.1.177

Port number：10000（目标板连接调试时需要一致）

选择 apply 启用，选择 Close 退出。



附 1.2.4 目标板端配置

1）打开超级终端，配置为 115200bps,8N1

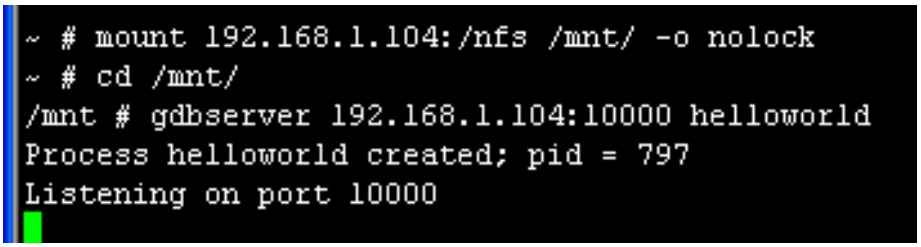
2）执行命令：mount 192.168.1.104:/nfs /mnt -o nolock

//挂载网络文件系统，其中 192.168.1.104 为虚拟机 IP 地址

3）执行命令：gdbserver 192.168.1.104:10000 helloworld

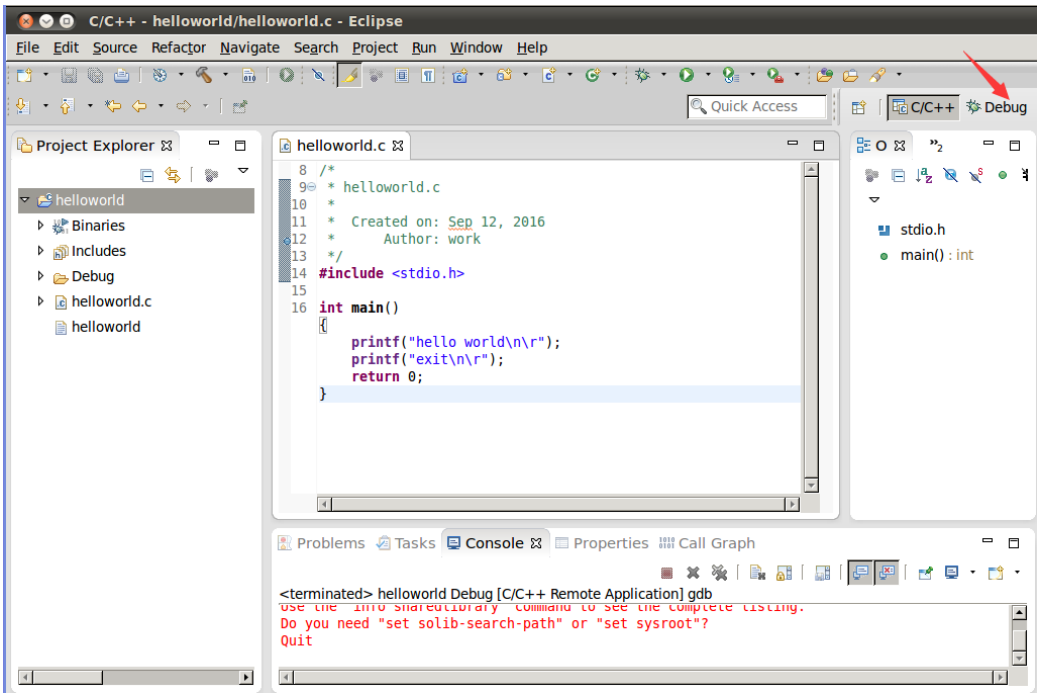
//10000 为端口号，需与 eclipse 配置时的端口号一致，helloworld 是需要调试的程序名称

4）执行完毕，gdbserver 已处于监听状态，如下图

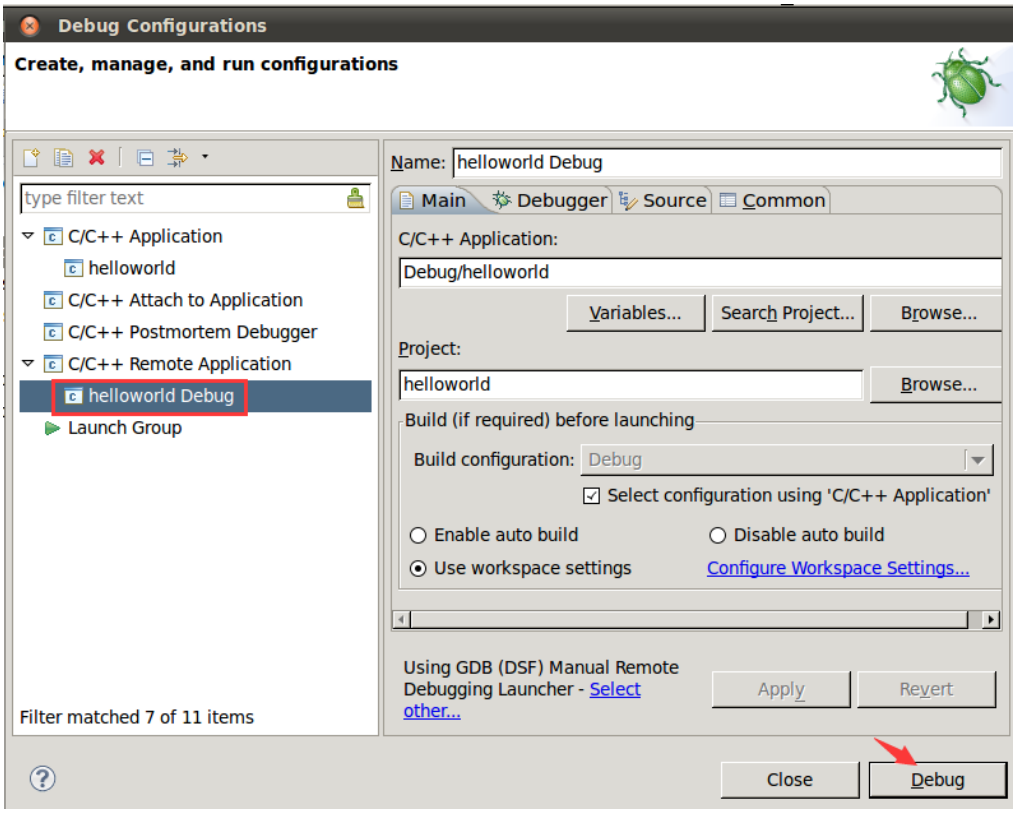


附 1.2.5 虚拟机端的 eclipse 端调试

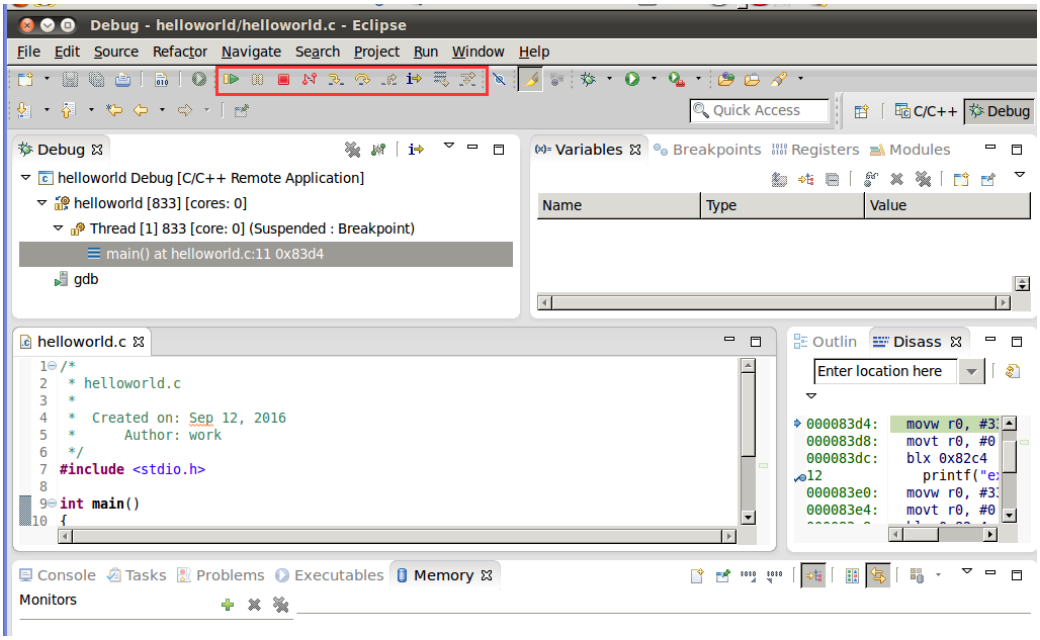
1）在 eclipse 开发环境里，切换到 Debug 模式



或者选择菜单 Run → Debug Configurations，进入配置窗口，在左侧窗口选中 helloworld Debug，点击右下角的 Debug，即可进行调试模式（在进入该调试模式之前，目标板需先打开 gdbserver 监听），如果有窗口弹出来，点击 Yes 即可



2）点击第一个图标为单步运行



六、驱动实例

在系统的/program 目录下有相应的脚本文件，可以进行一些简单的测试

1. RS485 接口驱动的使用

RS485-1 对应的驱动为 ttyS1；RS485-2 对应的驱动为 ttyS2

1.1 RS485 发送数据

附件：ssend.c

1.2 RS485 接收数据

附件：sread.c

2. GPRS 操作

GPRS 对应的驱动为 ttyS4

2.1 GPRS 收发数据

附件：at_u.c

2.2 GPRS 开机拨号

GPRS 拨号脚本为 dial-on.sh，如需要开机拨号，则可以把需要开机启动的脚本加入到脚本 startup.sh 里，如 startup.sh 不存在，则可以参考第五章的开机自启动步骤进行添加。

更多 AT 指令，请参考附件中的 M35F 目录的数据手册